

## AMENDMENTS TO THE CLAIMS

The following is a complete, marked up listing of revised claims with a status identifier in parentheses, underlined text indicating insertions, and strikethrough and/or double-bracketed text indicating deletions.

### List of Claims

1. (Previously Presented) A method for testing a service that provides interface functions having no user space interaction and that runs in a kernel space of an operating system, said method comprising:

generating a dynamically loadable kernel module (DLKM) containing a pseudo device driver having interface commands corresponding to the interface functions of the service to be tested, the generation being based upon function prototypes corresponding to the interface functions of the service to be tested;

loading the DLKM into the kernel space;

invoking the interface commands to exercise the interface functions to be tested; and

unloading the DLKM module.

2. (Original) A method in accordance with Claim 1 wherein invoking the interface commands further comprises loading a user library having user interfaces configured to test the kernel interfaces via the DLKM interface commands.

3. (Original) A method in accordance with Claim 2 wherein invoking the interface commands further comprises invoking the user interfaces of the user library via a program running in user space.

4. (Previously Presented) A method in accordance with Claim 1, wherein the generating of the DLKM includes:

reading a file of function prototypes for an interface function to be tested into a memory of a computing apparatus;

generating in the memory of the computer apparatus a representation of a conduit function from each function prototype, the conduit function having a user-space accessible invocation that activates an interface function corresponding to the function prototype from which the conduit function was generated; and

assembling the conduit function representations into the pseudo device driver.

5. (Original) A method in accordance with Claim 4 wherein invoking the interface commands further comprises invoking the conduit functions to invoke the interface commands.

6. (Original) A method in accordance with Claim 1 wherein unloading the DLKM module comprises removing the interface commands corresponding to the interface functions of the service to be tested, so that the interface commands are no longer available.

7. (Original) A method in accordance with Claim 1 wherein the interface commands of the DLKM module are accessible only to a privileged user or users, and not to another user or users.

8. (Original) A method in accordance with Claim 1 wherein invoking the interface commands to exercise the interface functions to be tested comprises opening the pseudo device driver, and invoking the interfaces of the device driver.

9. (Previously Presented) A machine readable medium having recorded thereon machine readable instructions, execution of which by a machine facilitates testing of a computing apparatus having a service with interface functions that have no user space interaction and that runs in a kernel space of an operating system, said machine readable medium comprising one or more code segments that represent instructions to:

generate a dynamically loadable kernel module (DLKM) containing a pseudo device driver having interface commands corresponding to the interface functions of the service to be tested, the generation being based upon function prototypes corresponding to the interface functions of the service to be tested;

load the DLKM into the kernel space;

invoke said interface commands to exercise the interface functions to be tested; and

unload said DLKM module.

10. (Previously Presented) A medium in accordance with Claim 9 wherein to invoke the interface commands, said machine readable medium further comprises one or more code segments that represent a user library having user interfaces, execution of which by the machine tests the kernel interfaces via said DLKM interface commands.

11. (Previously Presented) A medium in accordance with Claim 10 wherein to invoke said interface commands, said machine readable medium further comprises one or more code

segments that represent instructions to invoke said user interfaces of the user library via a program running in user space.

12. (Previously Presented) A medium in accordance with Claim 9 wherein said machine readable medium further comprises one or more code segments that represent instructions to:

read a file of function prototypes for an interface function to be tested into a memory of said computer apparatus;

generate in the memory of the computer apparatus a representation of a conduit function from each function prototype, said conduit function having a user-space accessible invocation that activates an interface function corresponding to the function prototype from which said conduit function was generated; and

assemble said conduit function representations into said pseudo device driver.

13. (Previously Presented) A medium in accordance with Claim 12 wherein to invoke the interface commands, said machine readable medium further comprises one or more code segments that represent instructions to invoke said conduit functions.

14. (Original) A medium in accordance with Claim 9 wherein the interface commands of the DLKM module are configured to be accessible only to a privileged user or users, and not to another user or users.

15. (Previously Presented) A computing apparatus comprising a processor and a storage device having recorded thereon a kernel of an operating system that includes a service

in kernel space that has interface functions without user space interaction, said computing apparatus configured to:

generate a dynamically loadable kernel module (DLKM) containing a pseudo device driver having interface commands corresponding to the interface functions of the service to be tested, the generation being based upon function prototypes corresponding to the interface functions of the service to be tested;

load the DLKM into the kernel space;

invoke said interface commands to exercise the interface functions to be tested; and

unload said DLKM module.

16. (Original) A computing apparatus in accordance with Claim 15 wherein to invoke the interface commands, said computing apparatus is further configured to load a user library having user interfaces that are configured to test the kernel interfaces via said DLKM interface commands.

17. (Original) A computing apparatus in accordance with Claim 16 wherein to invoke said interface commands, said computing apparatus is further configured to invoke said user interfaces of said user library via a program running in user space.

18. (Previously Presented) A computing apparatus in accordance with Claim 15 wherein to generate the DLKM, the computing apparatus is further configured to:

read a file of function prototypes for an interface function to be tested into a memory of said computing apparatus;

generate in said memory of said computing apparatus a representation of a conduit function from each function prototype, said conduit function having a user-space accessible

invocation that activates an interface function corresponding to the function prototype from which said conduit function was generated; and

assemble said conduit function representations into said pseudo device driver.

19. (Original) A computing apparatus in accordance with Claim 18 wherein to invoke said interface commands, said computing apparatus is further configured to invoke said conduit functions that invoke the interface commands.

20. (Original) A computing apparatus in accordance with Claim 15 wherein to unload said DLKM module, said computing apparatus is configured to remove said interface commands corresponding to the interface functions of the service to be tested, so that said interface commands are no longer available.

21. (Currently Amended) A method of generating a kernel module containing a pseudo device driver having interface commands corresponding to interface functions of a service to be tested, the service being arranged to run in a kernel space of an operating system and to provide one or more interface functions none of which exhibits user space interaction, said method comprising:

providing ~~the service with~~ one or more function prototypes corresponding to the one or more interface functions of the service to be tested; and

generating a dynamically loadable kernel module (DLKM) containing the pseudo device driver based upon the one or more function prototypes of the service to be tested.

22. (Currently Amended) A method in accordance with Claim 21, wherein:

the providing of ~~the service with the~~ one or more function prototypes includes the following,

aggregating the one or more function prototypes in a file associated with the service; and

the generating of the DLKM includes the following,

reading the file into a memory of a computing apparatus,

generating in the memory of the computer apparatus representations of one or more conduit functions for the one or more function prototypes, respectively, the one or more conduit functions having user-space accessible invocations that activate the corresponding interface functions, and

assembling the one or more conduit function representations into the pseudo device driver.

23. (Currently Amended) A machine readable medium having recorded thereon machine readable instructions, execution of which by a machine generates a kernel module containing a pseudo device driver having interface commands corresponding to interface functions of a service to be tested, the service being arranged to run in a kernel space of an operating system and to provide one or more interface functions none of which exhibits user space interaction, said machine readable medium comprising one or more code segments that represent instructions to:

provide ~~the service with~~ one or more function prototypes corresponding to the one or more interface functions of the service to be tested; and

generate a dynamically loadable kernel module (DLKM) containing the pseudo device driver based upon the one or more function prototypes of the service to be tested.

24. (Previously Presented) A medium in accordance with Claim 23 wherein:

to provide the one or more function prototypes, said machine readable medium further comprises one or more code segments representing instructions to do the following,

aggregate the one or more function prototypes in a file associated with the service; and

to generate the DLKM, said machine readable medium further comprises one or more code segments representing instructions to do the following,

read the file into a memory of a computing apparatus,

generate in the memory of the computer apparatus representations of one or more conduit functions for the one or more function prototypes, respectively, the one or more conduit functions having user-space accessible invocations that activate the corresponding interface functions, and

assemble the one or more conduit function representations into the pseudo device driver.

25. (Currently Amended) A computing apparatus comprising a processor and a storage device having recorded thereon a kernel of an operating system that includes a service in kernel space that has interface functions without user space interaction, said computing apparatus configured to:

provide ~~the service with~~ one or more function prototypes corresponding to the one or more interface functions of the service to be tested; and

generate a dynamically loadable kernel module (DLKM) containing the pseudo device driver based upon the one or more function prototypes of the service to be tested.



26. (Previously Presented) A computing apparatus in accordance with Claim 25 wherein:

to provide the one or more function prototypes, said computing apparatus further includes operability to do the following,

aggregate the one or more function prototypes in a file associated with the service; and

to generate the DLKM, said computing apparatus further includes operability to do the following,

read the file into a machine readable memory,

generate in the machine readable memory of the computer apparatus representations of one or more conduit functions for the one or more function prototypes, respectively, the one or more conduit functions having user-space accessible invocations that activate the corresponding interface functions, and

assemble the one or more conduit function representations into the pseudo device driver.

< Remainder of Page Intentionally Left Blank >